09-05-07

JFW

OIPE
SEP 04 2007
PATENT & TRADEMARK OFFICE

Attorney's Docket No. 042933/298965

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Nativadade Lobo — Confirmation No.: 5615
Appl. No.: 09/625,201
Filed: July 21, 2000
For: PULSE SHAPING WHICH COMPENSATES FOR COMPONENT DISTORTION

BOX ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMITTAL OF PRIORITY DOCUMENT

To complete the requirements of 35 U.S.C. § 119, enclosed is a certified copy of Great Britain priority Application No. 9801306.3, filed January 21, 1998.

Respectfully submitted,

Cory C. Davis

Cory C. Davis
Registration No. 59,932

Customer No. 00826
Alston & Bird LLP
Bank of America Plaza
101 South Tryon Street, Suite 4000
Charlotte, NC 28280-4000
Tel Charlotte Office (704) 444-1000
Fax Charlotte Office (704) 444-1111

"Express Mail" mailing label number EV521113148US
Date of Deposit September 4, 2007

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: BOX ISSUE FEE, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Tamara Stevens

Tamara Stevens

ellectual
perty
ice

For ... lty and Innovation

Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of e Comptroller-General, hereby certify that annexed hereto is a true copy of the documents originally filed in connection with patent application GB9801306.3 filed on 21 January 98.

ccordance with the Patents (Companies Re-registration) Rules 1982, if a company named his certificate and any accompanying documents has re-registered under the Companies 1980 with the same name as that with which it was registered immediately before re-stration save for the substitution as, or inclusion as, the last part of the name of the words lic limited company" or their equivalents in Welsh, references to the name of the pany in this certificate and any accompanying documents shall be treated as references to t ame with which it is so re-registered.

I cordance with the rules, the words "public limited company" may be replaced by p.l.c., p .L.C. or PLC.

R istration under the Companies Act does not constitute a new legal entity but merely su company to certain additional company law rules.

Signed

Dated 21 August 2007

Patents Form 1/77

Patents Act 1977
(Rule 16)

The Patent Office

9
23JAN98 E332520-1 D02916
P01/7700 25.00 - 9801306.3

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

THE PATENT OFFICE
G
21 JAN 1998
LONDON

The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference — PAT 98002 GB

2. Patent application number *(The Patent Office will fill in this part)* — 9801306.3 — 21 JAN 1998

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

NOKIA MOBILE PHONES LIMITED
KEILALAHDENTIE 4
02150 ESPOO
FINLAND

Patents ADP number *(if you know it)* — 59119 5004

If the applicant is a corporate body, give the country/state of its incorporation — FINLAND

4. Title of the invention

RECEIVER/MODULATOR

5. Name of your agent *(if you have one)* — MRS HELEN LOUISE HAWS

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

NOKIA MOBILE PHONES
PATENT DEPARTMENT
ST GEORGES COURT
ST GEORGES ROAD
CAMBERLEY
SURREY GU15 3QZ UK

Patents ADP number *(if you know it)* — 4105177001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

YES

I certify this to be a true copy.

G D Gent

Acting for Comptroller

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	30
Claim(s)	
Abstract	
Drawing(s)	IN WITH TEXT

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application.

Signature Date 21 January 1998

H L HAWS AGENT FOR THE APPLICANT

12. Name and daytime telephone number of person to contact in the United Kingdom

HELEN HAWS
01276 419346

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

In this paper we study Laurents paper.
Laurent shows that it is possible to construct Phase modulation by the superposition of amplitude modulated pulses.
The new idea :

We can separate the method of the superposition from the shape of the amplitude modulated pulses. We can then use different pulses of the same duration and /or fewer of them to construct modulation schemes that have desirable properties in the spectral and approximate to constant envelope schemes.

In particular we can use pulses with narrower bandwidth (but of the same support) to generate a modulation scheme that has a narrower bandwidth with a corresponding increase in the variation of the envelope of the signal .

In particular we can design mobile/ phones/ Systems with better power / bandwidth trade off considerations - and able to support higher number of ~~users per base stati~~ channels.

In this paper we effectly study Laurents paper.

We develop the modulating function that is used in GSM first to use as an example to demonstrate Laurent's idea.

$$\mathbf{T} := \frac{3}{812500}$$

$$\mathbf{BT} := 0.3$$

$$\sigma := \frac{\sqrt{\mathrm{Log}[2]}}{2\,\pi\,\mathrm{BT}}$$

$$\mathrm{h}[\mathrm{t\_}] := \frac{\mathrm{Exp}\left[-\frac{t^2}{2\,\sigma^2\,T^2}\right]}{\sqrt{2\,\pi}\,\sigma\,T}$$

Ideally we would and did define the effect of the convolution of h[t] by the formula below. However, this version of *Mathematica* gives an error.

$$\mathrm{h_{fil}}[\mathrm{t\_}] := \mathrm{Release}\left[\mathrm{Module}\left[\{\tau\},\ \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{\mathrm{h}[t-\tau]}{T}\,d\tau\right]\right]$$

We get round this by numerically fitting an interpolation function.

$$\mathrm{h_{fil}}[\mathrm{PulseWidth\_}][\mathrm{t\_}] := \mathrm{Module}\Big[\{x1,\ x2,\ x3\},\ x1 = (\mathrm{N}[\#1,\ 40]\&)\left[\mathrm{Table}\left[\left\{t1,\ \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{\mathrm{h}[t1-\tau]}{T}\,d\tau\right\},\ \left\{t1,\ -\frac{\mathrm{PulseWidth}}{2},\ \frac{\mathrm{PulseWidth}}{2},\ \frac{T}{20}\right\}\right]\right];\ x2 = \mathrm{Interpolation}[x1];\ x2[t]\Big]$$

```
Plot[h[t T], {t, -5, 5 }, PlotRange -> All]
```

- Graphics -

In this section we define $\phi_{L,t}$, the phase modulation function with a pulse width of length L at time t

$$\mathrm{ModulationIndex} := \frac{1}{2}$$

We use the definitions in the paper

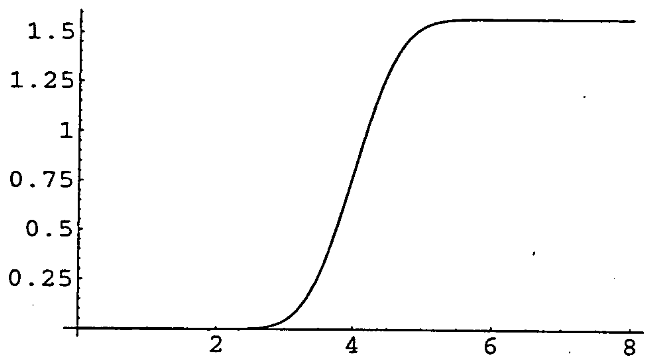
```
Φ := N[ModulationIndex π]

C := Cos[Φ];
S := Sin[Φ];
J := ( e^(j Φ) // Chop);
M := 2^(L-1);

Clear[PhaseAngle]

PhaseAngle[L_][t_] /; t ≤ 0 := 0

PhaseAngle[L_][t_] /; t ≥ L T := Φ
```

```
PhaseAngle[L_][t_] :=
 PhaseAngle[L][t_] = Module[{x1, x2, x3, x4, x5, x6}, x1 = h_fil[3 L T][t1 - (L T)/2];
   x2 = Table[{t2, Φ ∫_{-L T}^{t2} Evaluate[x1] ⅆt1}, {t2, 0, L T, T/100}]; Interpolation[x2][t]]

RuleDelayed::rhs : Pattern t_ appears on the right-hand side of
   rule φ_{L_,t_} :→ (φ_{L,t_} = Module[{x1, x2, x3, x4, x5, x6}, x1 = h_fil[3 L T][t1 - (L T)/2];
      x2 = Table[<<1>>]; Interpolation[x2][t]]).
```

When running the notebook we fix the value of L here and the rest of the graphs and functions use this value

```
L := 8;
PhaseFunction[t_] = φL,t;
phasepoints = Table[{t, PhaseFunction[t T]} // N, {t, 0, L , 1/40}];
ListPlot[phasepoints, PlotJoined -> True]
```

```
NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7
   recursive bisections in t1 near t1 = -3.68196×10^-7.

NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7
   recursive bisections in t1 near t1 = -7.4914×10^-7.

NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7
   recursive bisections in t1 near t1 = -5.43232×10^-7.

General::stop :
 Further output of NIntegrate::ncvb will be suppressed during this calculation.

NIntegrate::slwcon : Numerical integration converging too slowly; suspect singularity, value
   of the integration is 0, oscillatory integrand, or insufficient WorkingPrecision.
   If your integrand is oscillatory try using the option Method->Oscillatory in NIntegrate.

NIntegrate::slwcon : Numerical integration converging too slowly; suspect singularity, value
   of the integration is 0, oscillatory integrand, or insufficient WorkingPrecision.
   If your integrand is oscillatory try using the option Method->Oscillatory in NIntegrate.

NIntegrate::slwcon : Numerical integration converging too slowly; suspect singularity, value
   of the integration is 0, oscillatory integrand, or insufficient WorkingPrecision.
   If your integrand is oscillatory try using the option Method->Oscillatory in NIntegrate.

General::stop :
 Further output of NIntegrate::slwcon will be suppressed during this calculation.
```



- Graphics -

We can define an phase modulated signal to be

$S_t = E^{I \sum_{n=-\infty}^{\infty} \alpha_n \phi_{t-nT}}$

As can be seen from the graph ϕ_t is 0 for most of the time and is nearly Φ after LT . We denote by $\phi_{L,t}$ the pulse that is set to these values

i.e $\phi_{L,t} = 0 \quad t < 0$;

$\phi_{L,t} = \Phi \; t \geq L\,T$;

$\phi_{L,t} = \phi_t$ Otherwise

$$S_t = E^{I \sum_{n=-\infty}^{\infty} \alpha_n \phi_{L,t-nT}}$$

We now consider $t = N\,T + \tau$ where $0 \leq \tau < T$

$$S_t = E^{I \sum_{n=-\infty}^{\infty} \alpha_n \phi_{L,NT+\tau-nT}}$$

$$S_t = E^{I \sum_{n=-\infty}^{\infty} \alpha_n \phi_{L,\tau+(N-n)T}}$$

Making use of the fact that $\phi_{L,t} = 0 \quad t < 0$;

$$S_t = E^{I \sum_{n=-\infty}^{N} \alpha_n \phi_{L,\tau+(N-n)T}}$$

$$S_t = E^{I \sum_{n=-\infty}^{N-L} \alpha_n \phi_{L,\tau+(N-n)T}} \; E^{I \sum_{n=N-L+1}^{N} \alpha_n \phi_{L,\tau+(N-n)T}}$$

Making use of $\phi_{L,t} = \Phi \; t \geq L\,T$ we get

$$S_t = E^{I \sum_{n=-\infty}^{N-L} \alpha_n \Phi} \; E^{I \sum_{n=N-L+1}^{N} \alpha_n \phi_{L,\tau+(N-n)T}}$$

And using the notation for $J := e^{j \Phi}$

$$S_t = J^{\sum_{n=-\infty}^{N-L} \alpha_n} \; E^{I \sum_{n=N-L+1}^{N} \alpha_n \phi_{L,\tau+(N-n)T}}$$

Making the substitution $n = N - i$

$$S_t = J^{\sum_{n=-\infty}^{N-L} \alpha_n} \; E^{I \sum_{i=0}^{L-1} \alpha_{N-i} \phi_{L,\tau+iT}}$$

Express this as

$$S_t = J^{\sum_{n=-\infty}^{N-L} \alpha_n} \prod_{i=0}^{L-1} \left(E^{I \alpha_{N-i} \phi_{L,\tau+iT}}\right)$$

Since we have

```
(Sin[A - B]/Sin[A] + e^(-j A) Sin[B]/Sin[A] /. Sin[x_] -> (e^(j x) - e^(-j x))/(2 j)) // Simplify
```

$E^{-I\,B}$

And

$$\left(\frac{\texttt{Sin[A - B]}}{\texttt{Sin[A]}} + e^{jA}\,\frac{\texttt{Sin[B]}}{\texttt{Sin[A]}}\ \texttt{/. Sin[x\_]} \to \frac{(e^{jx} - e^{-jx})}{2\,j}\right)\ \texttt{// Simplify}$$

$E^{I\,B}$

We can make the following substitutions S := Sin[Φ]

$$S_t = J^{\sum_{n=-\infty}^{N-L}\alpha_n}\prod_{i=0}^{L-1}\left(\frac{\text{Sin}[\Phi - \phi_{L,\tau+iT}]}{S} + J^{\alpha_{N-i}}\,\frac{\text{Sin}[\phi_{L,\tau+iT}]}{S}\right)$$

We can make the substitution

```
ψ[L_, t_] /; 0 < t < L T := φ_{L,t}

ψ[L_, t_] /; 2 L T > t ≥ L T := Φ - φ_{L,t-L T}
```

We need to put this to avoid the function being extrapolated

```
ψ[L_, t_] /; ! (0 < t < L T) && ! (2 L T > t ≥ L T) := 0

Clear[tom, x1]

tom = (x1[t_] = ψ[L, t T]; Table[{t, x1[t]} // N, {t, 0, 2 L, 1/40}]);
ListPlot[tom, PlotJoined -> True]
```

Now define

```
Clear[LaurentS]

LaurentS[L_][n_][t_] := Sin[ψ[L, t + n T]] / S

Plot[LaurentS[L][2][t T], {t, -2, 10}]
```

```
- Graphics -
```

$$S_t = J^{\sum_{n=-\infty}^{N-L}\alpha_n}\prod_{i=0}^{L-1}\left(\frac{\text{Sin}[\Phi - \phi_{L,\tau+iT}]}{S} + J^{\alpha_{N-i}}\,\frac{\text{Sin}[\phi_{L,\tau+iT}]}{S}\right)$$

$$S_t = J^{\sum_{n=-\infty}^{N-L}\alpha_n}\prod_{i=0}^{L-1}\left(\frac{\text{Sin}[\psi_{\tau+iT+LT}]}{S} + J^{\alpha_{N-i}}\,\frac{\text{Sin}[\psi_{\tau+iT}]}{S}\right)$$

$$S_t = J^{\sum_{n=-\infty}^{N-L}\alpha_n}\prod_{i=0}^{L-1}\left(\frac{\text{Sin}[\psi_{t-NT+iT+LT}]}{S} + J^{\alpha_{N-i}}\,\frac{\text{Sin}[\psi_{t-NT+iT}]}{S}\right)$$

$$S_t = J^{\sum_{n=-\infty}^{N-L}\alpha_n}\prod_{i=0}^{L-1}(\text{LaurentS}[i + L - N][t] + J^{\alpha_{N-i}}\ \text{LaurentS}[i - N][t])$$

We try to get an insight into $\prod_{i=0}^{L-1}(\text{LaurentS}[i + L - N][t] + J^{\alpha_{N-i}}\ \text{LaurentS}[i - N][t])$

To this end we try defining

$$\texttt{Temp[L\_, N\_][t\_]} := \prod_{i=0}^{L-1}(\mathcal{S}[i + L - N][t] + J^{\alpha_{N-i}}\ \mathcal{S}[i - N][t])$$

```
(Temp[4, 0][t] // Expand // Chop) /. Complex[0, 1.] -> Complex[0, 1]
```

$I^{\alpha_{-3}+\alpha_{-2}+\alpha_{-1}+\alpha_0}\, S[0][t]\, S[1][t]\, S[2][t]\, S[3][t] + I^{\alpha_{-3}+\alpha_{-2}+\alpha_{-1}}\, S[1][t]\, S[2][t]\, S[3][t]\, S[4][t] +$
$I^{\alpha_{-3}+\alpha_{-2}+\alpha_0}\, S[0][t]\, S[2][t]\, S[3][t]\, S[5][t] + I^{\alpha_{-3}+\alpha_{-2}}\, S[2][t]\, S[3][t]\, S[4][t]\, S[5][t] +$
$I^{\alpha_{-3}+\alpha_{-1}+\alpha_0}\, S[0][t]\, S[1][t]\, S[3][t]\, S[6][t] + I^{\alpha_{-3}+\alpha_{-1}}\, S[1][t]\, S[3][t]\, S[4][t]\, S[6][t] +$
$I^{\alpha_{-3}+\alpha_0}\, S[0][t]\, S[3][t]\, S[5][t]\, S[6][t] + I^{\alpha_{-3}}\, S[3][t]\, S[4][t]\, S[5][t]\, S[6][t] +$
$I^{\alpha_{-2}+\alpha_{-1}+\alpha_0}\, S[0][t]\, S[1][t]\, S[2][t]\, S[7][t] + I^{\alpha_{-2}+\alpha_{-1}}\, S[1][t]\, S[2][t]\, S[4][t]\, S[7][t] +$
$I^{\alpha_{-2}+\alpha_0}\, S[0][t]\, S[2][t]\, S[5][t]\, S[7][t] + I^{\alpha_{-2}}\, S[2][t]\, S[4][t]\, S[5][t]\, S[7][t] +$
$I^{\alpha_{-1}+\alpha_0}\, S[0][t]\, S[1][t]\, S[6][t]\, S[7][t] + I^{\alpha_{-1}}\, S[1][t]\, S[4][t]\, S[6][t]\, S[7][t] +$
$I^{\alpha_0}\, S[0][t]\, S[5][t]\, S[6][t]\, S[7][t] + S[4][t]\, S[5][t]\, S[6][t]\, S[7][t]$

In the next section we define the

```
Notation[α_{LL_,K_,ii_} ⟺ AlphaKI[LL_][K_, ii_]]

Clear[AlphaKI]

AlphaKI[LL_][K_, i_] /; 0 < i < LL && 0 ≤ K < 2^(LL-1) :=
 Module[{x1, x2, x3, KNum}, x1 := (x2 = Mod[KNum, 2]; KNum = (KNum - x2)/2; x2);
  KNum = K; x3 = Table[x1, {ii, 0, LL - 1}]; x3[[i]]]
```

Check the function.

```
Sum[2^(ii - 1) AlphaKI[6][13, ii], {ii, 1, 6 - 1}]

13
```

We can now define the LaurentC functions.

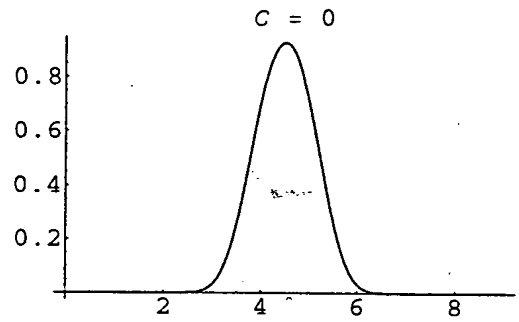
```
Clear[LaurentC]

General::spell1 : Possible spelling error: new symbol name "LaurentC" is similar
   to existing symbol "LaurentS".

LaurentC[L_][K_][t_] /; 0 ≤ K < 2^L :=
 LaurentS[L][0][t] ∏_{ii=1}^{L-1} LaurentS[L][ii + L AlphaKI[L][K, ii]][t]
```

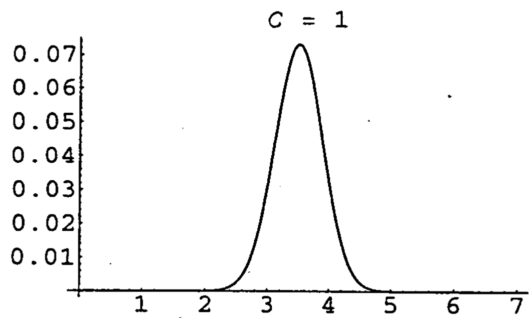
LaurentLK gives the support of the C functions

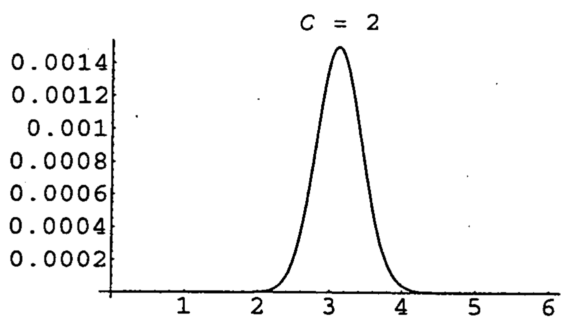
```
LaurentLK[L_][K_] :=
 Module[{x1}, x1 = Table[L (2 - AlphaKI[L][K, ii]) - ii, {ii, 1, L - 1}]; Min[x1]]

Table[LaurentLK[L][ii], {ii, 0, 2^(L-1) - 1}]

{9, 7, 6, 6, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}

CFunctionPlot[K_] := Module[{x1, x2}, x1[t_] = LaurentC[L][K][t T];
  x2 = Table[{t, x1[t ]}, {t, 0, LaurentLK[L][K] , 1 / 40 }]; ListPlot[x2,
   PlotJoined -> True, PlotLabel -> StringJoin["\n \n C = ", ToString[ii]],
   FormatType -> StandardForm, PlotRange -> All]]
```

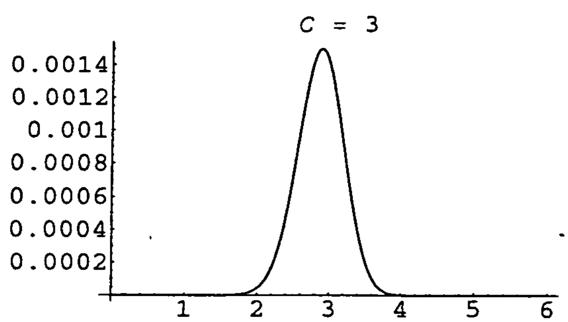
Plot at most the first 7 C functions

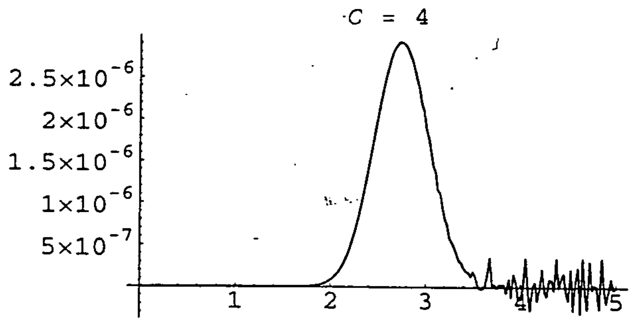
```
Table[CFunctionPlot[ii], {ii, 0, Min[6, 2^(L-1) - 1]}]
```

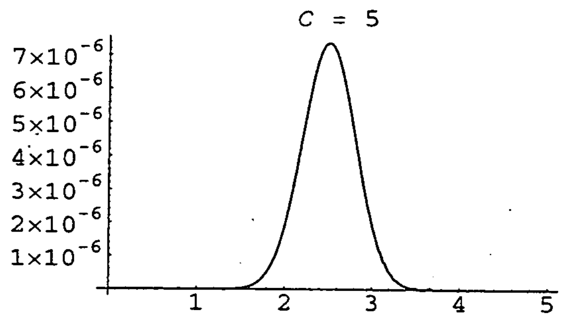

C = 0
0.8
0.6
0.4
0.2
2
4
6
8

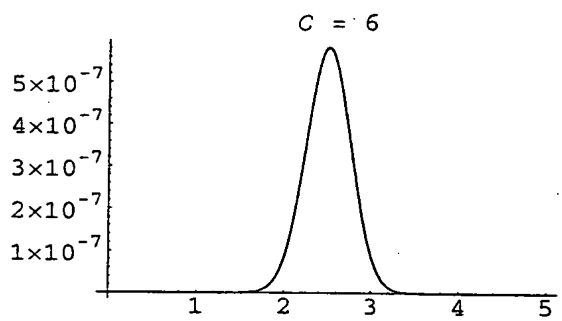

C = 1
0.07
0.06
0.05
0.04
0.03
0.02
0.01
1
2
3
4
5
6
7


C = 2
0.0014
0.0012
0.001
0.0008
0.0006
0.0004
0.0002
1
2
3
4
5
6


C = 3
0.0014
0.0012
0.001
0.0008
0.0006
0.0004
0.0002
1
2
3
4
5
6


C = 4
2.5×10^-6
2×10^-6
1.5×10^-6
1×10^-6
5×10^-7
1
2
3
4
5


C = 5
7×10^-6
6×10^-6
5×10^-6
4×10^-6
3×10^-6
2×10^-6
1×10^-6
1
2
3
4
5


C = 6
5×10^-7
4×10^-7
3×10^-7
2×10^-7
1×10^-7
1
2
3
4
5

{- Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -}

In this section we define the modulating function. We intend to build the modulator from first principles.

$$\sum_{n=-\infty}^{\infty}\sum_{K=0}^{M-1} J^{A_{K,n}}\, C_K[t-n\,T]$$

Change the order of summation

$$\sum_{K=0}^{M-1}\sum_{n=-\infty}^{\infty} J^{A_{K,n}}\, C_K[t-n\,T]$$

Given that we are interested at $t = N\,T + \tau$

$$\sum_{K=0}^{M-1}\sum_{n=-\infty}^{\infty} J^{A_{K,n}}\, C_K[\tau+(N-n)\,T]$$

Change variable $n \to n' + N$

$$\sum_{K=0}^{M-1}\sum_{n'=-\infty}^{\infty} J^{A_{K,n'+N}}\, C_K[\tau-n'\,T]$$

Notice that the support of C_K is positive only.

$$\sum_{K=0}^{M-1}\sum_{n'=-\infty}^{0} J^{A_{K,n'+N}}\, C_K[\tau-n'\,T]$$

Using LaurentLK as

$$\sum_{K=0}^{M-1}\sum_{n'=-\mathrm{LaurentLK}[L][K]+1}^{0} J^{A_{K,n'+N}}\, C_K[\tau-n'\,T]$$

Changing the sign of n

$$\sum_{K=0}^{M-1}\sum_{n=0}^{\mathrm{LaurentLK}[L][K]-1} J^{A_{K,N-n}}\, C_K[\tau+n\,T]$$

Let the value of $A_{0,0}$ be specified.

Then we can calculate the value of $A_{0,-1}$, $A_{0,-2}$, $A_{0,-3}$ using the following algorithm

$$A_{0,n-1} = A_{0,n} - \alpha_n$$

```
BitSeq = Table[1, {i, 1, 20}]
```

```
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
```

```
ANKInitialStateSetUp[L_][K_][InitBitSeq_, AccumulatedPhase_] :=
        Module[{x1, x2, x3, x4, x5, acuphase, initbitseq},
  initbitseq = InitBitSeq;
  acuphase = AccumulatedPhase;
   UpdateSeq :=
   Module[{},  x1 = acuphase - Sum[initbitseq[[i ]] AlphaKI[L][K, i], {i, 1, L - 1}];
    acuphase =  acuphase - First[initbitseq]; initbitseq = Rest[ initbitseq];  x1];
  Table[ UpdateSeq, {i, 1, LaurentLK[L][K]}]]
```

With the BitSeq we have we expect a decreasing sequence starting at 0.

```
ANKInitialStateSetUp[L][0][BitSeq, 0]
```

```
{0, -1, -2, -3, -4, -5, -6, -7, -8}
```

```
TestInit = Table[1, {i, 1, L}]
```

```
{1, 1, 1, 1, 1, 1, 1, 1}
```

```
AKN[L_][K_][{State_, AccumulatedPhase_}] :=
 AccumulatedPhase - Sum[State[[i + 1]] AlphaKI[L][K, i], {i, 1, L - 1}]

Options[Modulator] := {StartingQuadrant → 0, InitialState → Table[1, {i, 1, 20}],
  SamplingInterval → T / 32, NumberOfCurves → 4, ModulatingPulse → LaurentC}

BitSeq = Table[1, {i, 1, 20}]
```

```
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
```

```
Modulator[L_][BitSeq_, Opts___] :=
 Module[
  {x1, x2, x3, x4, x5, x6, state, AccumalatedPhase, seq, AKNState, Curves, Pulse},
x1 = SamplingInterval /. {Opts} /. Options[Modulator];
state = InitialState /. {Opts} /. Options[Modulator];
x3 = StartingQuadrant /. {Opts} /. Options[Modulator];
x4 = SamplingInterval /. {Opts} /. Options[Modulator];
Pulse = ModulatingPulse /. {Opts} /. Options[Modulator];
Curves = (NumberOfCurves /. {Opts} /. Options[Modulator] ) - 1;
AccumulatedPhase = x3 - BitSeq[[1]];
seq = BitSeq;
Table[AKNState[K] =
    ANKInitialStateSetUp[L][K][state, AccumulatedPhase], {K, 0, Curves}];
x5 := Module[{}, state = Join[{First[seq]}, Drop[state, -1]];
AccumulatedPhase = AccumulatedPhase + First[seq];
seq = Rest[seq];
Table[AKNState[K] = Join[{AKN[L][K][{state, AccumulatedPhase}]},
      Drop[AKNState[K], -1]], {K, 0, Curves}];
x6[τ_] = Sum[Sum[ (J)^AKNState[K][[i + 1]] Pulse[L][K][τ + i T],
      {i, 0, LaurentLK[L][K] - 1}], {K, 0, Curves}];
Table[x6[τ], {τ, 0, T - x4, x4}]];
 Table[x5, {kk , 1, Length[BitSeq]}] // Flatten]
```

```
General::spell1 :
 Possible spelling error: new symbol name "state" is similar to existing symbol "State".

General::spell1 : Possible spelling error: new symbol name "AccumalatedPhase" is
   similar to existing symbol "AccumulatedPhase".
```

```
RandomBitSeq = Table[Random[Integer, {0, 1}], {i, 1, 40}] // Map[# (-2) + 1&, #]&
```

```
{1, 1, 1, 1, 1, 1, 1, -1, 1, -1, 1, -1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, 1,
 1, 1, 1, 1, -1, -1, -1, -1, 1, -1, -1, 1, 1, -1, -1, 1, -1}
```

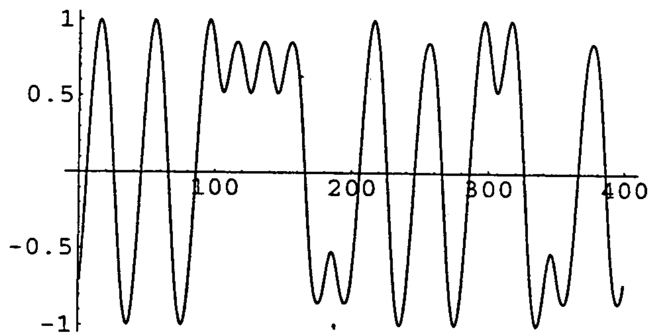
```
Modulator[L][BitSeq, SamplingInterval → T / 10];
```

```
Modulator[L][BitSeq, SamplingInterval → T / 10, NumberOfCurves -> 8];
```

```
Modulator[L][BitSeq, SamplingInterval → T / 10, NumberOfCurves -> 1];
```

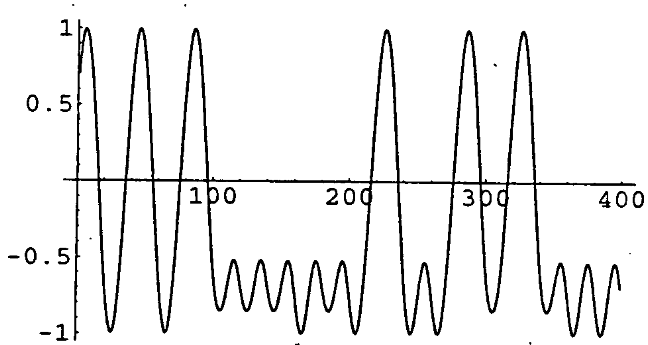
```
tom = Modulator[L][RandomBitSeq, SamplingInterval → T / 10, NumberOfCurves -> 2];
```

```
ListPlot[Im[tom], PlotJoined -> True]
```



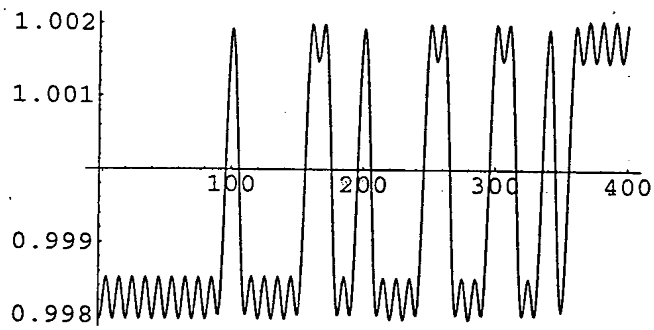
- Graphics -

```
ListPlot[Re[tom], PlotJoined -> True]
```



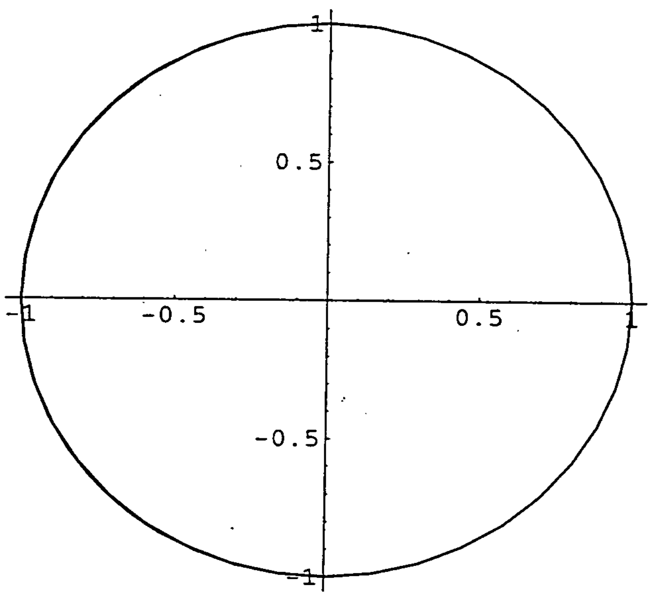
- Graphics -

```
ListPlot[Abs[tom], PlotJoined -> True]
```



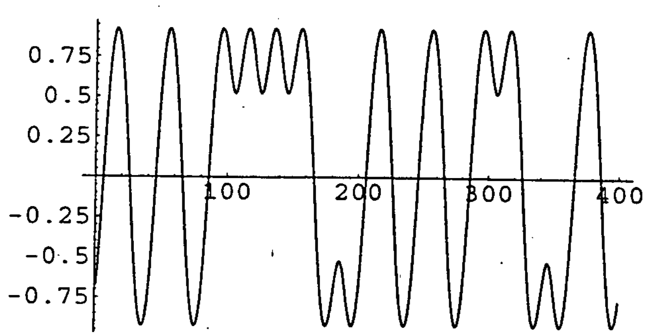
- Graphics -

```
{Re[tom], Im[tom]} // Transpose // ListPlot[#, PlotJoined -> True, AspectRatio -> 1]&
```



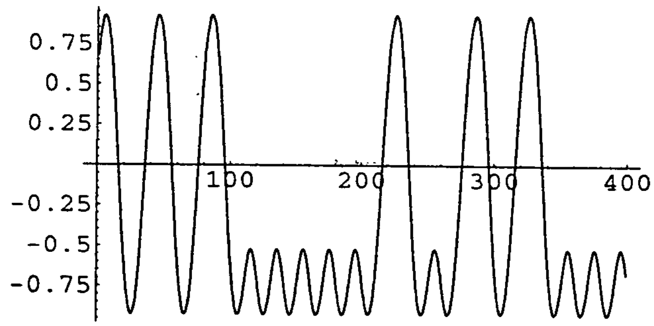
- Graphics -

```
tom = Modulator[L][RandomBitSeq, SamplingInterval → T / 10, NumberOfCurves -> 1];
ListPlot[Im[tom], PlotJoined -> True]
```



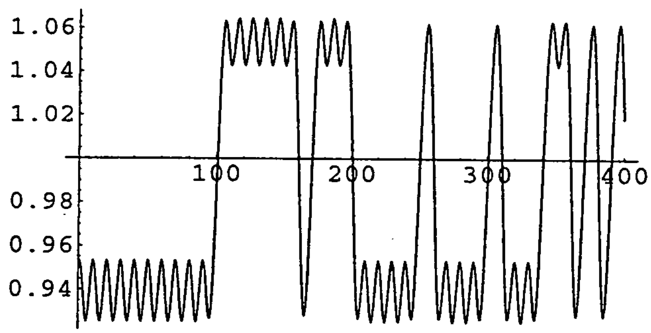
- Graphics -

```
ListPlot[Re[tom], PlotJoined -> True]
```



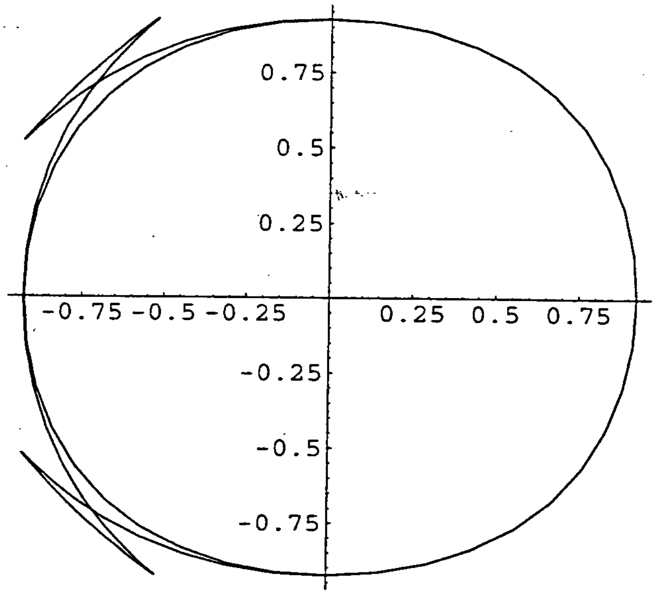
- Graphics -

```
ListPlot[Abs[tom], PlotJoined -> True]
```



- Graphics -

```
{Re[tom], Im[tom]} // Transpose // ListPlot[#, PlotJoined -> True, AspectRatio -> 1]&
```




```
- Graphics -
```

Calculating the bandwidth of the signal

```
x11 = Table[LaurentC[L][0][t T], {t, 0, L + 1 - 1/200, 1/200}];
((Drop[x11, -1] - Drop[x11, 1]) 200 )^2 /200 // Apply[Plus, #]&
1.29684
```

Here the pulse 6T is just stretched to 8T to see the performance

```
PhaseFunction[t_] = φ6,t;
phasepoints = Table[{t, PhaseFunction[t T]} // N, {t, 0, 6 , 1/40}];
ListPlot[phasepoints, PlotJoined -> True]
```

```
NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7
   recursive bisections in t1 near t1 = -1.91796×10^-7.

NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7
   recursive bisections in t1 near t1 = -5.52869×10^-7.

NIntegrate::slwcon : Numerical integration converging too slowly; suspect singularity, value
   of the integration is 0, oscillatory integrand, or insufficient WorkingPrecision.
   If your integrand is oscillatory try using the option Method->Oscillatory in NIntegrate.

NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7
   recursive bisections in t1 near t1 = -1.96696×10^-7.

General::stop :
 Further output of NIntegrate::ncvb will be suppressed during this calculation.

NIntegrate::slwcon : Numerical integration converging too slowly; suspect singularity, value
   of the integration is 0, oscillatory integrand, or insufficient WorkingPrecision.
   If your integrand is oscillatory try using the option Method->Oscillatory in NIntegrate.

NIntegrate::slwcon : Numerical integration converging too slowly; suspect singularity, value
   of the integration is 0, oscillatory integrand, or insufficient WorkingPrecision.
   If your integrand is oscillatory try using the option Method->Oscillatory in NIntegrate.

General::stop :
 Further output of NIntegrate::slwcon will be suppressed during this calculation.
```

- Graphics -

```
TestPulse[8][0][t_] = LaurentC[6][0][t 7/9]
```

$$1.\ \mathrm{Sin}\left[\psi\left[6, \frac{3}{812500} + \frac{7t}{9}\right]\right] \mathrm{Sin}\left[\psi\left[6, \frac{3}{406250} + \frac{7t}{9}\right]\right] \mathrm{Sin}\left[\psi\left[6, \frac{9}{812500} + \frac{7t}{9}\right]\right] \mathrm{Sin}\left[\psi\left[6, \frac{3}{203125} + \frac{7t}{9}\right]\right] \mathrm{Sin}\left[\psi\left[6, \frac{3}{162500} + \frac{7t}{9}\right]\right] \mathrm{Sin}\left[\psi\left[6, \frac{7t}{9}\right]\right]$$

```
Plot[TestPulse[8][0][t], {t, 0, 9 T}, PlotRange -> All]
```

- Graphics -

```
RandomBitSeq = Table[Random[Integer, {0, 1}], {i, 1, 100}] // Map[# (-2) + 1&, #]&
```

```
{1, 1, -1, -1, -1, 1, 1, -1, 1, -1, 1, -1, 1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, 1, 1, 1,
 -1, 1, -1, -1, 1, -1, -1, 1, 1, 1, 1, -1, 1, -1, 1, -1, 1, -1, -1, 1, 1, 1, 1, -1,
 1, -1, 1, -1, 1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1, -1, -1, -1, 1, -1,
 1, 1, 1, -1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, 1, -1}
```

```
tom = Modulator[L][RandomBitSeq, SamplingInterval → T/10, NumberOfCurves -> 1,
  ModulatingPulse → TestPulse];
```

```
ListPlot[Im[tom], PlotJoined -> True]
```

- Graphics -

```
ListPlot[Re[tom], PlotJoined -> True]
```

- Graphics -

```
ListPlot[Abs[tom], PlotJoined -> True]
```

- Graphics -

```
{Re[tom], Im[tom]} // Transpose // ListPlot[#, PlotJoined -> True, AspectRatio -> 1]&
```

- Graphics -

In this section we consider filtering the pulse

```
Needs["bessel`"]

fil = BesselFilter[8][LowPass3dB[1 / T 2 Pi 0.3]][FrequencyResponse][s];

Plot[20 Log[10, Evaluate[(fil /. s → 2 Pi I f / T) // Abs]], {f, 0, 2},
 GridLines -> Automatic, AxesLabel -> {"1/T", "dB"}]
```

- Graphics -

```
Res[t_] = InverseLaplaceTransform[fil, s, t]
```

$2027025 ((9.36812 + 0. I) E^{-897172. t} Cos[139301. t] - (13.7944 + 0. I) E^{-835672. t} Cos[420044. t] + (4.73111 + 0. I) E^{-701358. t} Cos[708768. t] - (0.304883 + 0. I) E^{-455818. t} Cos[1.02016 \times 10^6 t] + (29.1264 + 0. I) E^{-897172. t} Sin[139301. t] - (9.89398 + 0. I) E^{-835672. t} Sin[420044. t] - (0.321205 + 0. I) E^{-701358. t} Sin[708768. t] + (0.375154 + 0. I) E^{-455818. t} Sin[1.02016 \times 10^6 t])$

```
Plot[Res[t T], {t, 0, 6}, PlotRange -> All]
```

- Graphics -

```
filTaps = Table[Res[t T], {t, 0, 6 , 1 / 20}] // Chop;
```

We estimate the group delay to be 1.8T

```
Length[filTaps]
```

121

```
pulse = Table[LaurentC[8][0][t T], {t, 0, 9, 1 / 20}];
```

```
General::spell1 :
 Possible spelling error: new symbol name "pulse" is similar to existing symbol "Pulse".
```

```
pulse2 = Table[LaurentC[8][1][t T], {t, 0, 7, 1 / 20}];

Needs["Convolve`"]
```

```
FilteredPulse = T / 20 MapConvolve[pulse, filTaps] // Take[#, {34, Length[#] }]&
```

{1.73731×10^{-15}, 5.35887×10^{-15}, 1.60413×10^{-14}, 4.66278×10^{-14}, 1.31688×10^{-13},
3.61578×10^{-13}, 9.65729×10^{-13}, 2.51044×10^{-12}, 6.35507×10^{-12}, 1.56747×10^{-11},
3.76889×10^{-11}, 8.83867×10^{-11}, 2.02275×10^{-10}, 4.51958×10^{-10}, 9.86455×10^{-10},
2.10426×10^{-9}, 4.38922×10^{-9}, 8.95697×10^{-9}, 1.78914×10^{-8}, 3.4999×10^{-8}, 6.70842×10^{-8},
1.26054×10^{-7}, 2.32319×10^{-7}, 4.20166×10^{-7}, 7.46062×10^{-7}, 1.30124×10^{-6},
2.23035×10^{-6}, 3.75853×10^{-6}, 6.23×10^{-6}, 0.0000101618, 0.0000163174, 0.0000258049,
0.0000402068, 0.0000611746, 0.000093496, 0.00013964, 0.000205786, 0.000299337,
0.000429924, 0.000609895, 0.000854853, 0.00118424, 0.00162196, 0.00219697,
0.0029439, 0.00390362, 0.00512368, 0.00665873, 0.00857068, 0.0109288, 0.0138093,
0.0172954, 0.021476, 0.0264451, 0.0323001, 0.0391407, 0.0470666, 0.0561754,
0.0665603, 0.0783079, 0.0914948, 0.106186, 0.12243, 0.140261, 0.159691, 0.180711,
0.20329, 0.22737, 0.25287, 0.279682, 0.307673, 0.336686, 0.366539, 0.397029,
0.427934, 0.459013, 0.490013, 0.520668, 0.550708, 0.579856, 0.607837, 0.634379,
0.65922, 0.682107, 0.702805, 0.721098, 0.73679, 0.749713, 0.759727, 0.76672,
0.770616, 0.771369, 0.768969, 0.763442, 0.754846, 0.743274, 0.728852, 0.711736,
0.692108, 0.670178, 0.646176, 0.62035, 0.592964, 0.56429, 0.534607, 0.504193,
0.473326, 0.442277, 0.411305, 0.380656, 0.350558, 0.321221, 0.29283, 0.26555,
0.239519, 0.214852, 0.191636, 0.169936, 0.149791, 0.13122, 0.114217, 0.0987613,
0.0848118, 0.0723143, 0.0612016, 0.0513967, 0.0428146, 0.035365, 0.0289542,
0.023487, 0.0188687, 0.0150065, 0.0118109, 0.00919671, 0.00708412, 0.0053992,
0.00407439, 0.00304877, 0.00226811, 0.0016848, 0.00125761, 0.000951417, 0.000736739,
0.000589309, 0.000489543, 0.000422011, 0.000374899, 0.000339478, 0.0003096,
0.000281223, 0.000251974, 0.000220764, 0.000187441, 0.000152497, 0.000116824,
0.0000815146, 0.0000477044, 0.0000164566, -0.0000113198, -0.0000349206,
-0.0000538729, -0.0000679391, -0.0000771062, -0.0000815622, -0.0000816651,
-0.0000779073, -0.000070877, -0.000061222, -0.0000496142, -0.0000367186,
-0.0000231668, -9.53584×10^{-6}, 3.66817×10^{-6}, 0.0000160205, 0.0000271838,
0.0000369093, 0.0000450345, 0.0000514769, 0.0000562268, 0.0000593363, 0.000060909,
0.0000610882, 0.0000600456, 0.0000579702, 0.0000550589, 0.0000515076, 0.0000475036,
0.0000432203, 0.000038813, 0.0000344153, 0.0000301387, 0.0000260716,
0.0000222801, 0.0000188096, 0.000015687, 0.0000129228, 0.000010514,
8.4469×10^{-6}, 6.69967×10^{-6}, 5.24481×10^{-6}, 4.05145×10^{-6}, 3.08723×10^{-6},
2.3199×10^{-6}, 1.71859×10^{-6}, 1.25466×10^{-6}, 9.0236×10^{-7}, 6.39109×10^{-7},
4.45602×10^{-7}, 3.05722×10^{-7}, 2.06318×10^{-7}, 1.36898×10^{-7}, 8.92721×10^{-8},
5.71866×10^{-8}, 3.59686×10^{-8}, 2.22017×10^{-8}, 1.34418×10^{-8}, 7.97812×10^{-9},
4.63953×10^{-9}, 2.64191×10^{-9}, 1.4722×10^{-9}, 8.02333×10^{-10}, 4.27388×10^{-10},
2.2234×10^{-10}, 1.12886×10^{-10}, 5.59276×10^{-11}, 2.70717×10^{-11}, 1.28211×10^{-11},
5.92683×10^{-12}, 2.65259×10^{-12}, 1.14156×10^{-12}, 4.73562×10^{-13}, 1.94329×10^{-13},
7.95416×10^{-14}, 3.14517×10^{-14}, 1.23089×10^{-14}, 4.73302×10^{-15}, 1.66595×10^{-15},
4.65283×10^{-16}, 1.93314×10^{-16}, 9.87738×10^{-17}, 2.31472×10^{-18}, -1.50823×10^{-17},
-1.01228×10^{-17}, -6.49877×10^{-18}, -2.08455×10^{-18}, 1.80177×10^{-19}, 1.70062×10^{-19},
1.9917×10^{-19}, -1.14008×10^{-19}, -1.66045×10^{-20}, 3.7877×10^{-21}, 1.47752×10^{-21},
1.94753×10^{-21}, 3.72298×10^{-22}, 2.02535×10^{-22}, 1.36007×10^{-22}, 6.72514×10^{-24},
7.07296×10^{-23}, 6.84862×10^{-23}, 3.68916×10^{-23}, 2.6013×10^{-24}, -1.51251×10^{-24},
-5.51867×10^{-25}, 7.52915×10^{-26}, 5.81584×10^{-25}, 1.75371×10^{-25}, 1.22392×10^{-26}, 0}

```
FilteredPulse2 = T / 20 MapConvolve[pulse2, filTaps] // Take[#, {50, Length[#] }]&
```

{1.27972×10^-7, 2.18306×10^-7, 3.66972×10^-7, 6.0806×10^-7, 9.93427×10^-7, 1.60076×10^-6, 2.54471×10^-6, 3.99199×10^-6, 6.18149×10^-6, 9.45059×10^-6, 0.0000142688, 0.0000212804, 0.0000313562, 0.0000456572, 0.0000657079, 0.0000934812, 0.000131493, 0.000182902, 0.000251616, 0.000342391, 0.000460925, 0.000613932, 0.000809185, 0.00105553, 0.00136282, 0.00174186, 0.00220418, 0.00276183, 0.00342705, 0.00421186, 0.00512763, 0.00618453, 0.00739101, 0.00875319, 0.0102744, 0.0119544, 0.0137895, 0.0157714, 0.0178876, 0.0201213, 0.0224509, 0.024851, 0.0272921, 0.0297419, 0.0321652, 0.0345256, 0.0367857, 0.0389085, 0.0408581, 0.0426011, 0.044107, 0.0453496, 0.0463073, 0.0469636, 0.0473081, 0.0473359, 0.0470483, 0.0464525, 0.0455611, 0.0443919, 0.0429671, 0.0413131, 0.039459, 0.0374364, 0.0352785, 0.0330189, 0.0306913, 0.0283283, 0.0259611, 0.0236189, 0.021328, 0.019112, 0.0169912, 0.0149823, 0.0130987, 0.0113504, 0.0097438, 0.00828249, 0.00696697, 0.00579518, 0.00476282, 0.00386365, 0.0030899, 0.00243259, 0.00188191, 0.00142752, 0.00105885, 0.000765388, 0.000536896, 0.000363597, 0.000236336, 0.000146695, 0.0000870728, 0.0000507313, 0.0000318118, 0.0000253258, 0.0000271231, 0.0000338422, 0.0000428466, 0.0000521516, 0.0000603451, 0.0000665053, 0.0000701193, 0.0000710042, 0.0000692325, 0.0000650647, 0.0000588885, 0.0000511665, 0.0000423925, 0.0000330554, 0.0000236122, 0.0000144671, 5.95924×10^-6, -1.64542×10^-6, -8.15591×10^-6, -0.0000134543, -0.0000174907, -0.0000202751, -0.0000218687, -0.0000223734, -0.0000219217, -0.000020666, -0.0000187698, -0.0000163988, -0.0000137138, -0.0000108643, -7.98457×10^-6, -5.18972×10^-6, -2.57416×10^-6, -2.1065×10^-7, 1.84943×10^-6, 3.57494×10^-6, 4.95319×10^-6, 5.98756×10^-6, 6.69487×10^-6, 7.10243×10^-6, 7.24522×10^-6, 7.16306×10^-6, 6.89804×10^-6, 6.49223×10^-6, 5.98586×10^-6, 5.41573×10^-6, 4.81423×10^-6, 4.20863×10^-6, 3.6208×10^-6, 3.06728×10^-6, 2.55958×10^-6, 2.10471×10^-6, 1.70583×10^-6, 1.36297×10^-6, 1.07376×10^-6, 8.34162×10^-7, 6.3908×10^-7, 4.82884×10^-7, 3.59854×10^-7, 2.64486×10^-7, 1.91715×10^-7, 1.37044×10^-7, 9.65984×10^-8, 6.71324×10^-8, 4.59913×10^-8, 3.10539×10^-8, 2.06612×10^-8, 1.3542×10^-8, 8.74155×10^-9, 5.55593×10^-9, 3.47564×10^-9, 2.13956×10^-9, 1.29613×10^-9, 7.73129×10^-10, 4.54101×10^-10, 2.62031×10^-10, 1.47888×10^-10, 8.15113×10^-11, 4.40884×10^-11, 2.37341×10^-11, 1.26865×10^-11, 6.58958×10^-12, 3.34992×10^-12, 1.6239×10^-12, 7.01037×10^-13, 2.33036×10^-13, 7.75051×10^-14, -3.22249×10^-15, -8.12875×10^-14, -7.89371×10^-14, -5.29088×10^-14, -3.14297×10^-14, -8.21475×10^-15, 4.12557×10^-15, 1.9906×10^-15, 1.82068×10^-15, -4.36969×10^-15, 4.67143×10^-16, 7.65162×10^-16, 5.67707×10^-16, 4.93342×10^-16, 7.52562×10^-17, 7.75606×10^-17, 4.50401×10^-17, -2.18469×10^-17, 1.39012×10^-16, 1.92164×10^-16, 6.31946×10^-17, -6.72328×10^-18, 1.24704×10^-17, 1.42063×10^-17, 7.33335×10^-18, 1.13717×10^-19, 3.90929×10^-19, 1.30285×10^-19, 0}

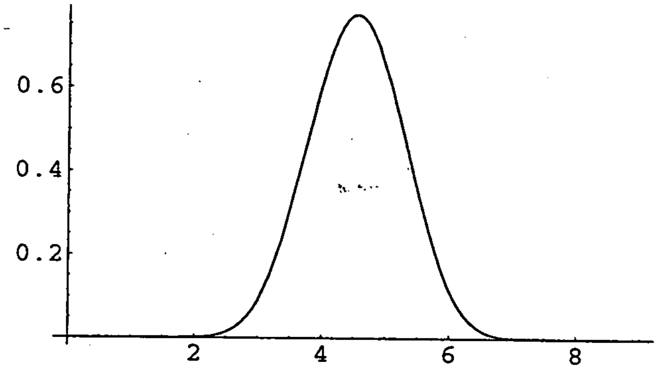
```
FiltPulse[8][0] = {(Table[t T, {t, 0, 20, 1/20}] // N) //
     Take[#, Length[FilteredPulse]]&, FilteredPulse} // Transpose //
  Interpolation
```

InterpolatingFunction[{{0, 0.0000492923}}, <>]

```
FiltPulse[8][1] = {(Table[t T, {t, 0, 20, 1/20}] // N) //
     Take[#, Length[FilteredPulse2]]&, FilteredPulse2} // Transpose //
  Interpolation
```

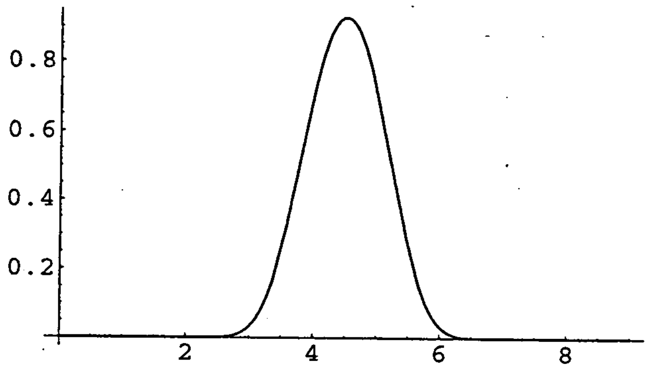
InterpolatingFunction[{{0, 0.0000389538}}, <>]

```
Plot[FiltPulse[8][0][t T], {t, 0, 9}, PlotRange -> All]
```



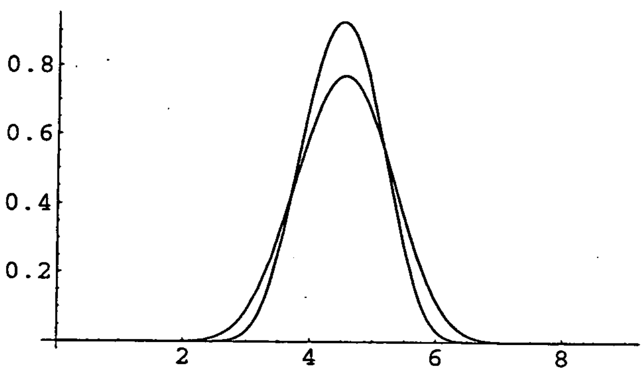
- Graphics -

```
Plot[LaurentC[8][0][t T], {t, 0, 9}, PlotRange -> All]
```



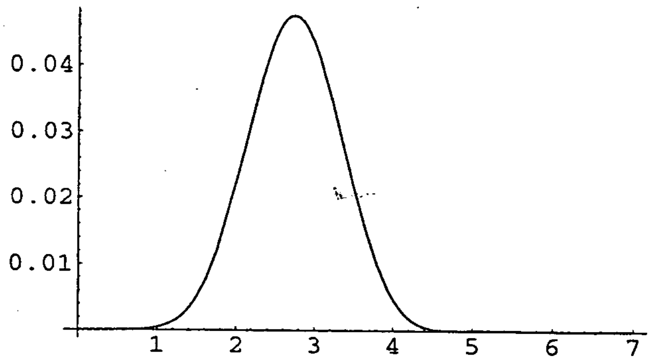
- Graphics -

```
Show[%, %%]
```



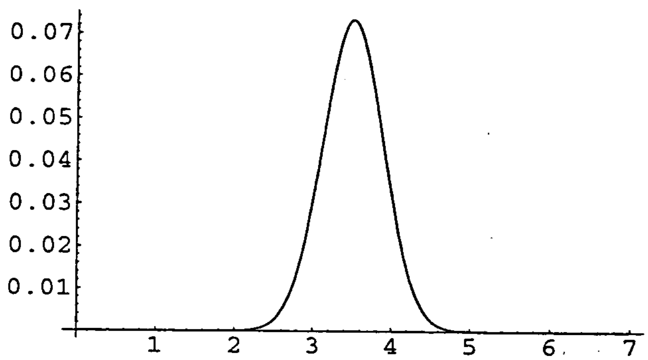
- Graphics -

```
Plot[FiltPulse[8][1][t T], {t, 0, 7}, PlotRange -> All]
```



- Graphics -

```
Plot[LaurentC[8][1][t T], {t, 0, 7}, PlotRange -> All]
```



- Graphics -

```
BandWidth[TestPulse_, Supp_] :=
 Module[{x11}, x11 = Table[TestPulse[t T], {t, 0, Supp - 1 / 200, 1 / 200}];
((Drop[x11, -1] - Drop[x11, 1]) 200 ) ^2 / 200 // Apply[Plus, #]&]
```

```
BandWidth[FiltPulse[8][0], 9]
```

0.711659

```
BandWidth[LaurentC[8][0], 9]
```

1.29684

```
BandWidth[FiltPulse[8][1], 7]
```

0.00338446

```
tom = Modulator[L][Table[1, {i, 1, 100}], SamplingInterval → T / 10,
  NumberOfCurves -> 2, ModulatingPulse → FiltPulse];
```

```
tom = Modulator[L][RandomBitSeq, SamplingInterval → T / 10, NumberOfCurves -> 2,
  ModulatingPulse → FiltPulse];
```

```
ListPlot[Im[tom], PlotJoined -> True]
```

- Graphics -

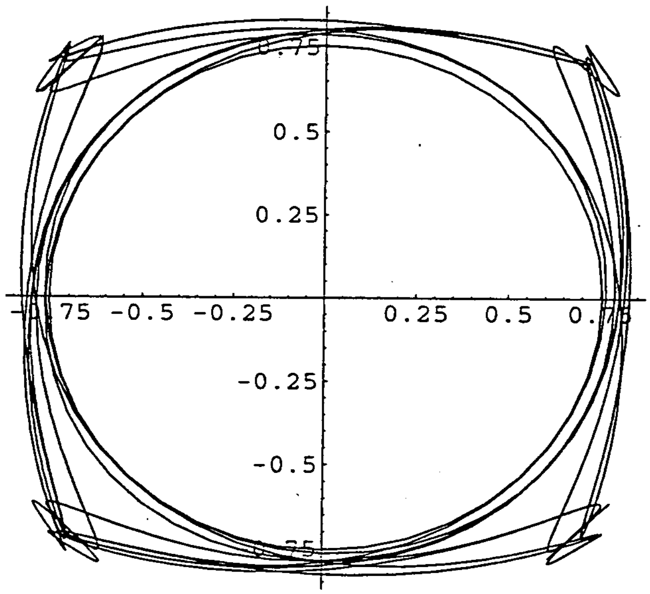
```
ListPlot[Re[tom], PlotJoined -> True]
```

- Graphics -

```
ListPlot[Abs[tom], PlotJoined -> True]
```

- Graphics -

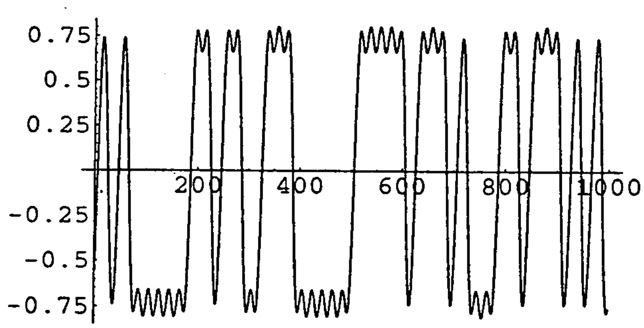
```
{Re[tom], Im[tom]} // Transpose // ListPlot[#, PlotJoined -> True, AspectRatio -> 1]&
```



- Graphics -

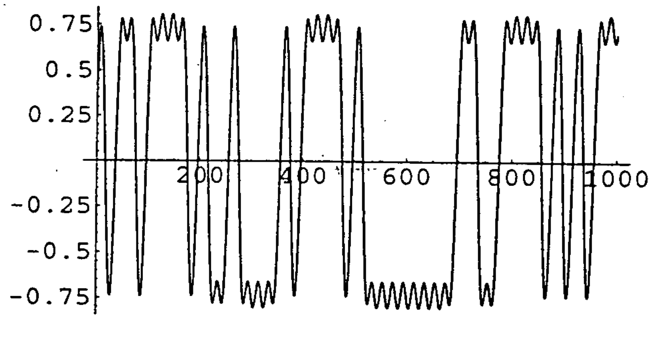
```
tom = Modulator[L][RandomBitSeq, SamplingInterval → T / 10, NumberOfCurves -> 1,
  ModulatingPulse → FiltPulse];
```

```
ListPlot[Im[tom], PlotJoined -> True]
```



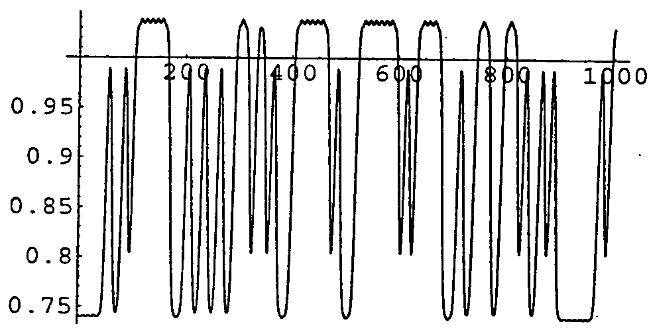
- Graphics -

```
ListPlot[Re[tom], PlotJoined -> True]
```



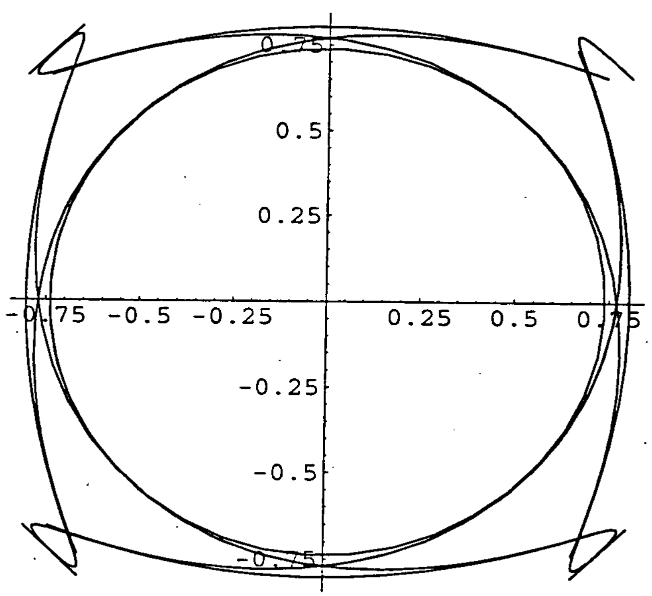
- Graphics -

```
ListPlot[Abs[tom], PlotJoined -> True]
```



- Graphics -

```
{Re[tom], Im[tom]} // Transpose // ListPlot[#, PlotJoined -> True, AspectRatio -> 1]&
```



- Graphics -

```
x11 = Table[TestPulse[6][0][t T], {t, 0, 7 - 1/200, 1/200}];
```

```
((Drop[x11, -1] - Drop[x11, 1]) 200 ) ^2 / 200 // Apply[Plus, #]&
```

0.92632

```
TestPulse[6][0][t_] = LaurentC[4][0][t 5/7 ]
```

$$1.\ \text{Sin}\left[\psi\left[4,\ \frac{3}{812500}+\frac{5\,t}{7}\right]\right]\ \text{Sin}\left[\psi\left[4,\ \frac{3}{406250}+\frac{5\,t}{7}\right]\right]\ \text{Sin}\left[\psi\left[4,\ \frac{9}{812500}+\frac{5\,t}{7}\right]\right]\ \text{Sin}\left[\psi\left[4,\ \frac{5\,t}{7}\right]\right]$$

```
L := 5;
PhaseFunction[t_] = φL,t;
phasepoints = Table[{t, PhaseFunction[t T]} // N, {t, 0, L , 1/40}];
ListPlot[phasepoints, PlotJoined -> True]
```

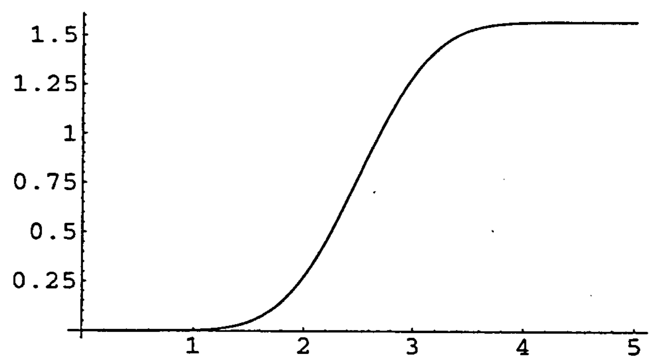
NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7 recursive bisections in t1 near t1 = -1.48053×10^{-6}.

NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7 recursive bisections in t1 near t1 = -1.12255×10^{-6}.

NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7 recursive bisections in t1 near t1 = -1.08808×10^{-6}.

General::stop : Further output of NIntegrate::ncvb will be suppressed during this calculation.

NIntegrate::slwcon : Numerical integration converging too slowly; suspect singularity, value of the integration is 0, oscillatory integrand, or insufficient WorkingPrecision. If your integrand is oscillatory try using the option Method->Oscillatory in NIntegrate.



- Graphics -

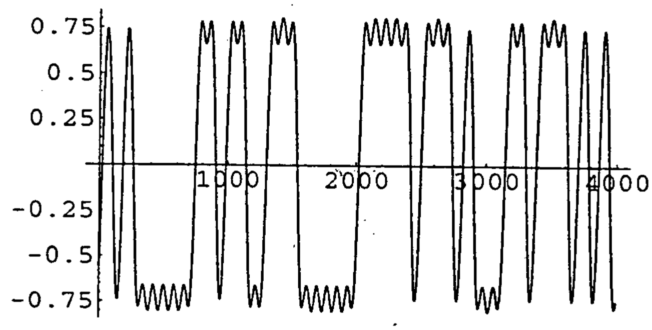
```
L := 6
```

```
TestPulse2[6][0][t_] = LaurentC[4][0][t 6/7 ]
```

$$1.\ \text{Sin}\left[\psi\left[4,\ \frac{3}{812500}+\frac{6\,t}{7}\right]\right]\ \text{Sin}\left[\psi\left[4,\ \frac{3}{406250}+\frac{6\,t}{7}\right]\right]\ \text{Sin}\left[\psi\left[4,\ \frac{9}{812500}+\frac{6\,t}{7}\right]\right]\ \text{Sin}\left[\psi\left[4,\ \frac{6\,t}{7}\right]\right]$$

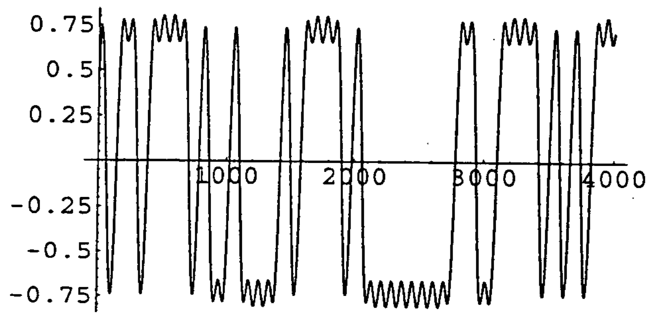
```
tom = Modulator[L][RandomBitSeq, SamplingInterval → T/40, NumberOfCurves -> 1,
  ModulatingPulse → FiltPulse];

Save["ModulatorData.m", {T, L, RandomBitSeq, FiltPulse, tom}]
```

```
ListPlot[Im[tom], PlotJoined -> True]
```



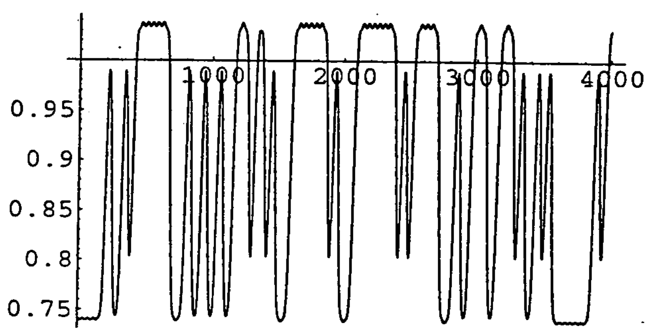
- Graphics -

```
ListPlot[Re[tom], PlotJoined -> True]
```



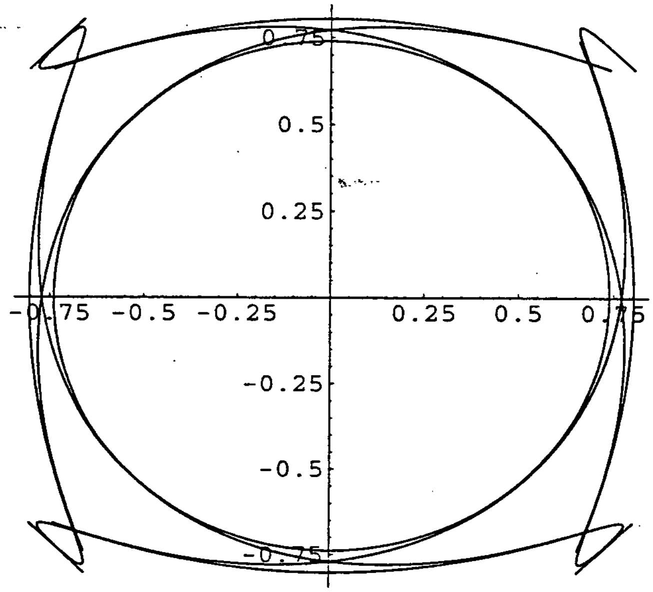
- Graphics -

```
ListPlot[Abs[tom], PlotJoined -> True]
```



- Graphics -

```
{Re[tom], Im[tom]} // Transpose // ListPlot[#, PlotJoined -> True, AspectRatio -> 1]&
```



- Graphics -

```
x11 = Table[TestPulse2[6][0][t T], {t, 0, 7 - 1/200, 1/200}];
```

BeginPackage["LaurentFunctions`"]

T::usage = "This is the symbol period"

BT::usage = "This is the usual product"

h::usage = "This is the raw gaussian pulse"

Φ::usage = "This denotes modulation index Pi"

ψ::usage = "ψ[L,t] is Laurents function"

hFiltered::usage = "This is the filtered gaussian pulse"

PhaseAngle::usage = "This function takes some time to calculate. The following code will draw a graph of the functionPhaseFunction[t_] =N[$\phi_{L,t}$];
phasepoints =Table[{t,PhaseFunction[t T]}//N,{t,0,L ,1/40}];
ListPlot[phasepoints,PlotJoined -> True]"

S ::usage = "S = Sin[Φ]"

J::usage = "J = $e^{j\Phi}$"

C::usage = "C = Cos[Φ]"

M::usage = "M = 2^{L-1}"

ModulationIndex::usage = "ModulationIndex = h"

LaurentS::usage = "LaurentS[L][n][t] = Sin[ψ[L,t + n T]]/ S"

LaurentC::usage = "LaurentC[L][K][t] is Laurents $C_{K,t}$"

AlphaKI::usage = "AlphaKI[LL][K,i] is Laurents $\alpha_{K,i}$"

LaurentLK::usage = "LaurentLK[L][K] gives the support of $C_{K,t}$"

The start of Modulator Definitions

ANKInitialStateSetUp::usage =
"ANKInitialStateSetUp[L][K][InitBitSeq,AccumulatedPhase] sets up the sequence of prior states of $A_{K,N}$ that the modulator went thgough to get to the constelation point specified by AccumulatedPhase which is really $A_{0,0}$"

AKN::usage = "AKN[L][K][{State,AccumulatedPhase}] defines $A_{K,N}$ in terms of $A_{0,N}$"

ModulatingPulse::usage =
"The Pulse is assumed to have the following structure Pulse[L][K][t]"

NumberOfCurves::usage = "The number of pulses used by the modulator"

SamplingInterval::usage =
"SamplingInterval is the interval between samples of the output of the modulator"

InitialState::usage = "InitialState is the set of bits that are assumed to be present before i.e {a_{-1},a_{-2}, ...}"

StartingQuadrant::usage = "StartingQuadrant = $A_{0,-1}$and is a number"

Modulator::usage = "Modulator[L][BitSeq,Opts] assumes the following default options StartingQuadrant → 0,InitialState → Table[1,{i, 1,20}],SamplingInterval → T/32,NumberOfCurves → 4,ModulatingPulse→ LaurentC . The Pulse is assumed to have the following structure Pulse[L][K][t]"

Start of the Receiver Functions

```
FiltPulse::usage = "The default pulse and is called by FiltPulse[L][K][t]"

SyncSample::usage = "Given that the sampling interval is T/32,then sync
   sample has rang -16 to 16 and this moves the point used to demodulate "

Receiver::usage = " Receiver[L][InputSeq,Opts] assumes the following
   default options {StartingQuadrant→0,InitialState→{1,1,1,1,1,1,1,1,
   1,1,1,1,1,1,1,1,1,1,1,1},SamplingInterval -> T/32,ModulatingPulse →
   FiltPulse,SyncSample -> 0,NumberOfCurves -> 2  "

ReceiverProper::usage = "ReceiverProper[L][ StartingQuadrant, InitialState,
   SamplingInterval,ModulatingPulse,SyncSample,NumberOfCurves,InputSeq]
   is called by Receiver after all the options have been resolved"

Begin["`Private`"]
```

$$\sigma := \frac{\sqrt{\mathrm{Log}[2]}}{2\,\pi\,\mathrm{BT}}$$

$$\mathrm{h}[\mathrm{t\_}] := \frac{\mathrm{Exp}\left[-\frac{t^2}{2\,\sigma^2\,T^2}\right]}{\sqrt{2\,\pi}\,\sigma\,T}$$

Ideally we would and did define the effect of the convolution of h[t] by the formula below. However, this version of *Mathematica* gives an error.

$$\mathrm{h_{fil}}[\mathrm{t\_}] := \mathrm{Release}\left[\mathrm{Module}\left[\{\tau\},\ \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{\mathrm{h}[t-\tau]}{T}\, d\tau\right]\right]$$

$$\mathrm{hFiltered}[\mathrm{PulseWidth\_}][\mathrm{t\_}] := \mathrm{Module}\Big[\{x1,\ x2,\ x3\},\ x1 = (\mathrm{N}[\#1,\ 40]\&)\left[\mathrm{Table}\left[\left\{t1,\ \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{\mathrm{h}[t1-\tau]}{T}\, d\tau\right\},\ \left\{t1,\ -\frac{\mathrm{PulseWidth}}{2},\ \frac{\mathrm{PulseWidth}}{2},\ \frac{T}{20}\right\}\right]\right];$$
$$x2 = \mathrm{Interpolation}[x1];\ x2[t]\Big]$$

$$\Phi := \mathrm{N}[\mathrm{ModulationIndex}\,\pi]$$

$$C := \mathrm{Cos}[\Phi];$$
$$S := \mathrm{Sin}[\Phi];$$
$$J := (e^{j\,\Phi}\ //\ \mathrm{Chop});$$
$$M := 2^{L-1};$$

$$\mathrm{PhaseAngle}[\mathrm{L\_}][\mathrm{t\_}]\ /;\ t \le 0 := 0$$

$$\mathrm{PhaseAngle}[\mathrm{L\_}][\mathrm{t\_}]\ /;\ t \ge L\,T := \Phi$$

$$\mathrm{PhaseAngle}[\mathrm{L\_}][\mathrm{t\_}] :=$$
$$\mathrm{PhaseAngle}[L][\mathrm{t\_}] = \mathrm{Module}\left[\{x1,\ x2,\ x3,\ x4,\ x5,\ x6\},\ x1 = \mathrm{hFiltered}[3\,L\,T]\left[t1 - \frac{L\,T}{2}\right];\right.$$
$$\left. x2 = \mathrm{Table}\left[\left\{t2,\ \Phi \int_{-L\,T}^{t2} \mathrm{Evaluate}[x1]\, dt1\right\},\ \left\{t2,\ 0,\ L\,T,\ \frac{T}{100}\right\}\right];\ \mathrm{Interpolation}[x2][t]\right]$$

$$\psi[\mathrm{L\_},\ \mathrm{t\_}]\ /;\ 0 < t < L\,T := \mathrm{PhaseAngle}[L][t]$$

$$\psi[\mathrm{L\_},\ \mathrm{t\_}]\ /;\ 2\,L\,T > t \ge L\,T := \Phi - \mathrm{PhaseAngle}[L][t - L\,T]$$

We need to put this to avoid the function being extrapolated

$$\psi[\mathrm{L\_},\ \mathrm{t\_}]\ /;\ !\ (0 < t < L\,T)\ \&\&\ !\ (2\,L\,T > t \ge L\,T) := 0$$

$$\mathrm{LaurentS}[\mathrm{L\_}][\mathrm{n\_}][\mathrm{t\_}] := \mathrm{Sin}[\psi[L,\ t + n\,T]]\ /\ S$$

```
AlphaKI[LL_][K_, i_] /; (0 < i < LL) && (0 <= K < 2^(LL-1)) :=
 Module[{x1, x2, x3, KNum}, x1 := (x2 = Mod[KNum, 2]; KNum = (KNum - x2)/2; x2);
  KNum = K; x3 = Table[x1, {ii, 0, LL - 1}]; x3[[i]]]

LaurentLK[L_][K_] :=
 Module[{x1}, x1 = Table[L (2 - AlphaKI[L][K, ii]) - ii, {ii, 1, L - 1}]; Min[x1]]

LaurentC[L_][K_][t_] /; 0 <= K < 2^L :=
 LaurentS[L][0][t] Product[LaurentS[L][ii + L AlphaKI[L][K, ii]][t], {ii, 1, L-1}]
```

The start of the modulator function

```
ANKInitialStateSetUp[L_][K_][InitBitSeq_, AccumulatedPhase_] :=
        Module[{x1, x2, x3, x4, x5, acuphase, initbitseq},
  initbitseq = InitBitSeq;
  acuphase = AccumulatedPhase;
   UpdateSeq :=
   Module[{},  x1 = acuphase - Sum[initbitseq[[i ]] AlphaKI[L][K, i], {i, 1, L - 1}];
    acuphase =  acuphase - First[initbitseq]; initbitseq = Rest[ initbitseq];  x1];
  Table[ UpdateSeq, {i, 1, LaurentLK[L][K]}]]

AKN[L_][K_][{State_, AccumulatedPhase_}] :=
 AccumulatedPhase - Sum[State[[i + 1]] AlphaKI[L][K, i], {i, 1, L - 1}]

Options[Modulator] := {StartingQuadrant → 0, InitialState → Table[1, {i, 1, 20}],
  SamplingInterval → T / 32, NumberOfCurves → 4, ModulatingPulse → LaurentC}

Modulator[L_][BitSeq_, Opts___] :=
 Module[
  {x1, x2, x3, x4, x5, x6, state, AccumalatedPhase, seq, AKNState, Curves, Pulse},
x1 = SamplingInterval /. {Opts} /. Options[Modulator];
state = InitialState /. {Opts} /. Options[Modulator];
x3 = StartingQuadrant /. {Opts} /. Options[Modulator];
x4 = SamplingInterval /. {Opts} /. Options[Modulator];
Pulse = ModulatingPulse /. {Opts} /. Options[Modulator];
Curves = (NumberOfCurves /. {Opts} /. Options[Modulator] ) - 1;
AccumulatedPhase = x3 ;
seq = BitSeq;
Table[
   AKNState[K] = ANKInitialStateSetUp[L][K][state, AccumulatedPhase], {K, 0, Curves}];
x5 := Module[{}, state = Join[{First[seq]}, Drop[state, -1]];
AccumulatedPhase = AccumulatedPhase + First[seq];
seq = Rest[seq];
Table[AKNState[K] = Join[{AKN[L][K][{state, AccumulatedPhase}]},
      Drop[AKNState[K], -1]], {K, 0, Curves}];
x6[τ_] = Sum[Sum[ (J)^AKNState[K][[i + 1]] Pulse[L][K][τ + i T],
      {i, 0, LaurentLK[L][K] - 1}], {K, 0, Curves}];
Table[x6[τ], {τ, 0, T - x4, x4}]];
 Table[x5, {kk , 1, Length[BitSeq]}] // Flatten]

Options[Receiver] := {StartingQuadrant → 0,
  InitialState → {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}, SamplingInter
  ModulatingPulse → FiltPulse, SyncSample -> 0, NumberOfCurves -> 2};

Receiver[L_][InputSeq_, Opts___] :=
  Module[{x1, x2, x3, x4, x5, x6},
  x1 = StartingQuadrant /. {Opts} /. Options[Receiver];
  x2 = InitialState /. {Opts} /. Options[Receiver];
  x3 = SamplingInterval /. {Opts} /. Options[Receiver];
  x4 = ModulatingPulse /. {Opts} /. Options[Receiver];
  x5 = SyncSample /. {Opts} /. Options[Receiver];
  x6 = NumberOfCurves /. {Opts} /. Options[Receiver];
ReceiverProper[L][x1, x2, x3, x4, x5, x6, InputSeq]]
```

```
ReceiverProper[L_][StartingQuadrant_, InitialState_, SamplingInterval_,
  ModulatingPulse_, SyncSample_, NumberOfCurves_, InputSeq_] :=
     Module[{x1, sgn, ReceivedSeq,  ExpectedValue, seq, ReceiveNext, D, 𝒜},
  x1 = T / SamplingInterval;
  ReceivedSeq = Partition[InputSeq, x1] // Transpose // #[[SyncSample + 1]]&;
  ExpectedValue =
   ModulatingPulse[L][0][(LaurentLK[L][0] / 2) T + SyncSample SamplingInterval];
  𝒜 = StartingQuadrant;
  sgn = 0;
  D = {};
  seq = ReceivedSeq;
  ReceiveNext :=
  Module[{x1, x2},
    x1 = ( (-1)^sgn J^𝒜 First[seq]) // Im;
   If[Abs[x1 - ExpectedValue] <= Abs[x1 + ExpectedValue],
     D = Join[ D, {1}];  𝒜 = 𝒜 + 1,  D = Join[ D, {-1}];  𝒜 = 𝒜 - 1 ];
    seq = Rest[seq]; sgn = Mod[sgn + 1, 2]];
 Table[ReceiveNext, {i, 1, Length[ReceivedSeq]}];
  D]

End[]

EndPackage[]
```